Understanding Memory Management on iPhone

Session 416

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

Understanding Memory Management on iPhone

Dave Myszewski, Charles Srisuwananukorn
iPhone Performance

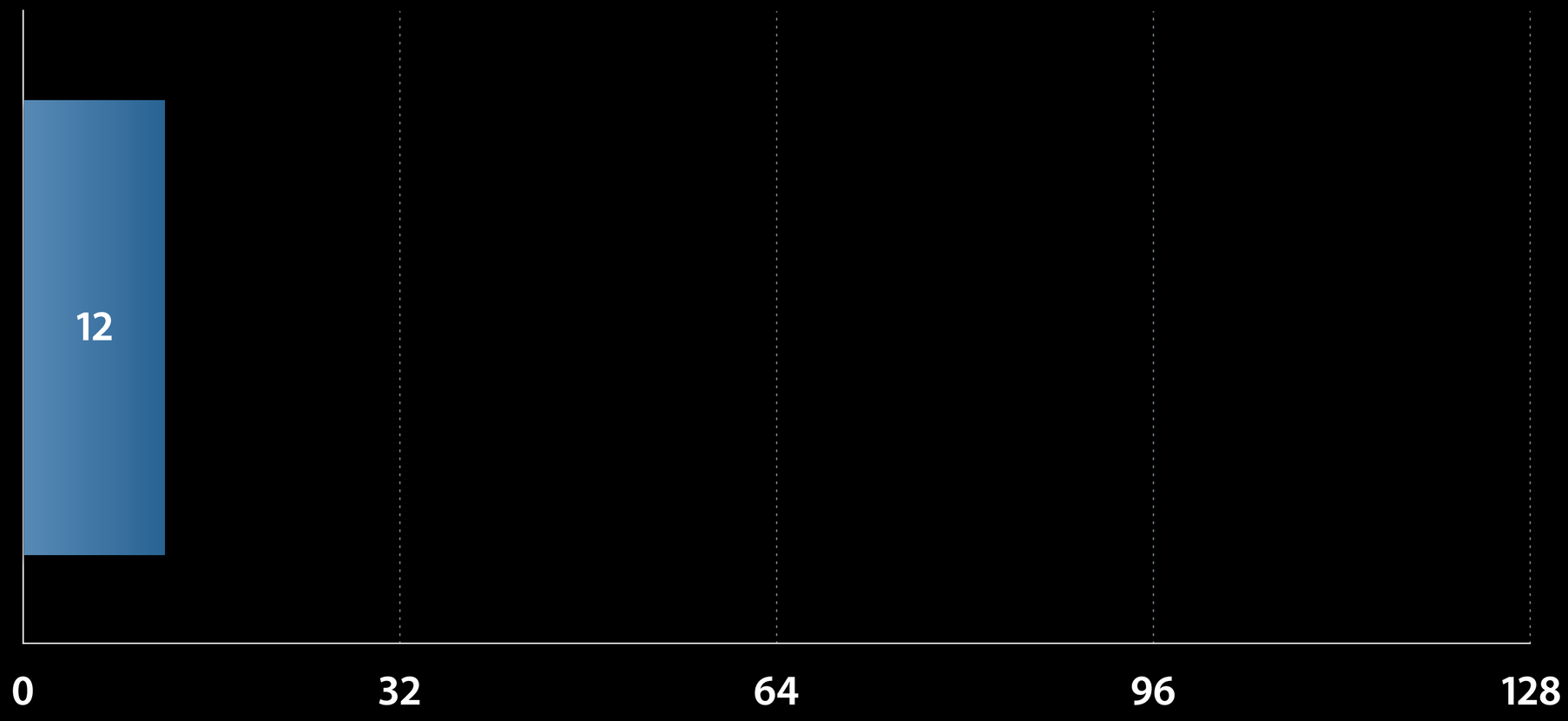
What You'll Learn

- Memory architecture on iPhone
- Memory tools
- Image and layer memory
- Responding to low memory warnings
- Autorelease pools
- Understanding low memory logs

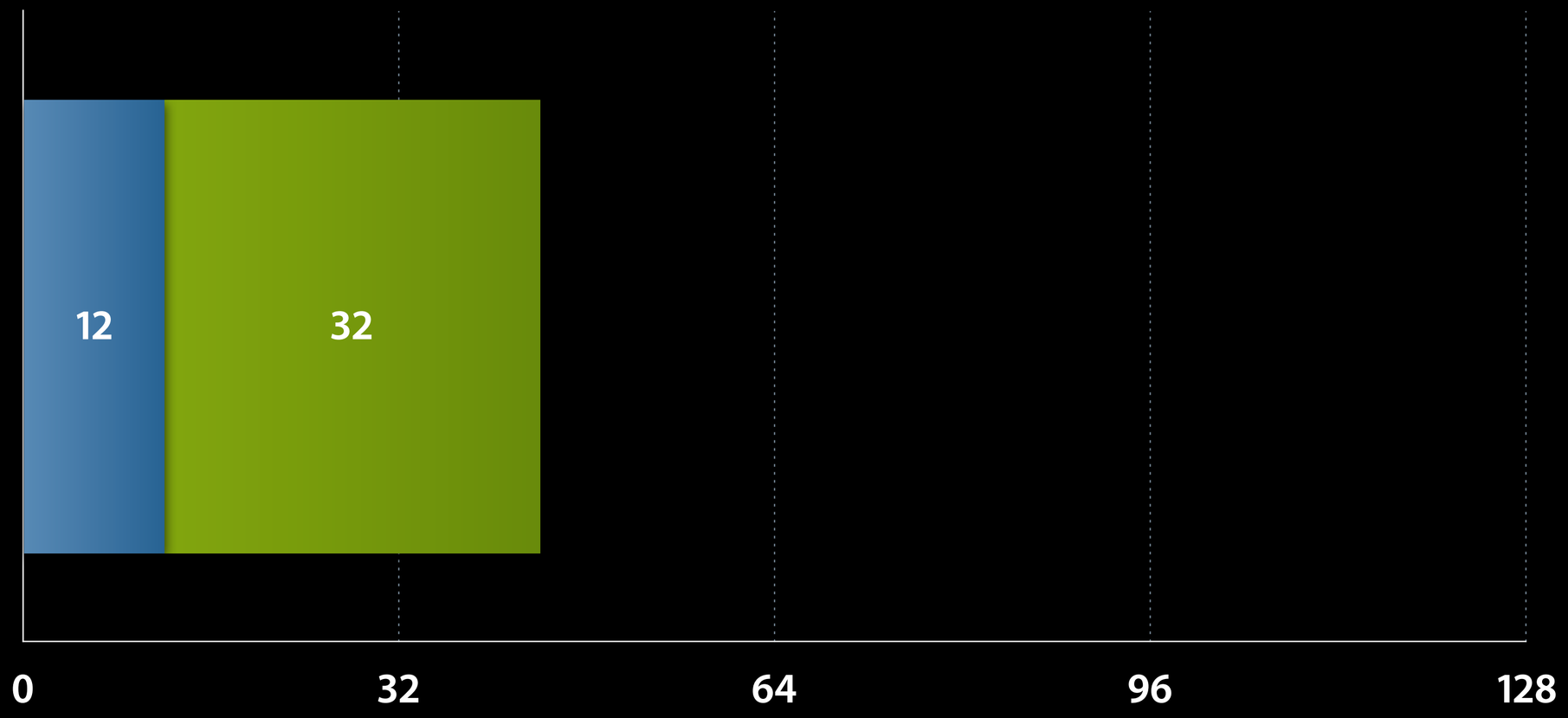
Not a Desktop OS

- Limited memory
- Virtual memory, but no swap file
- Low memory notifications

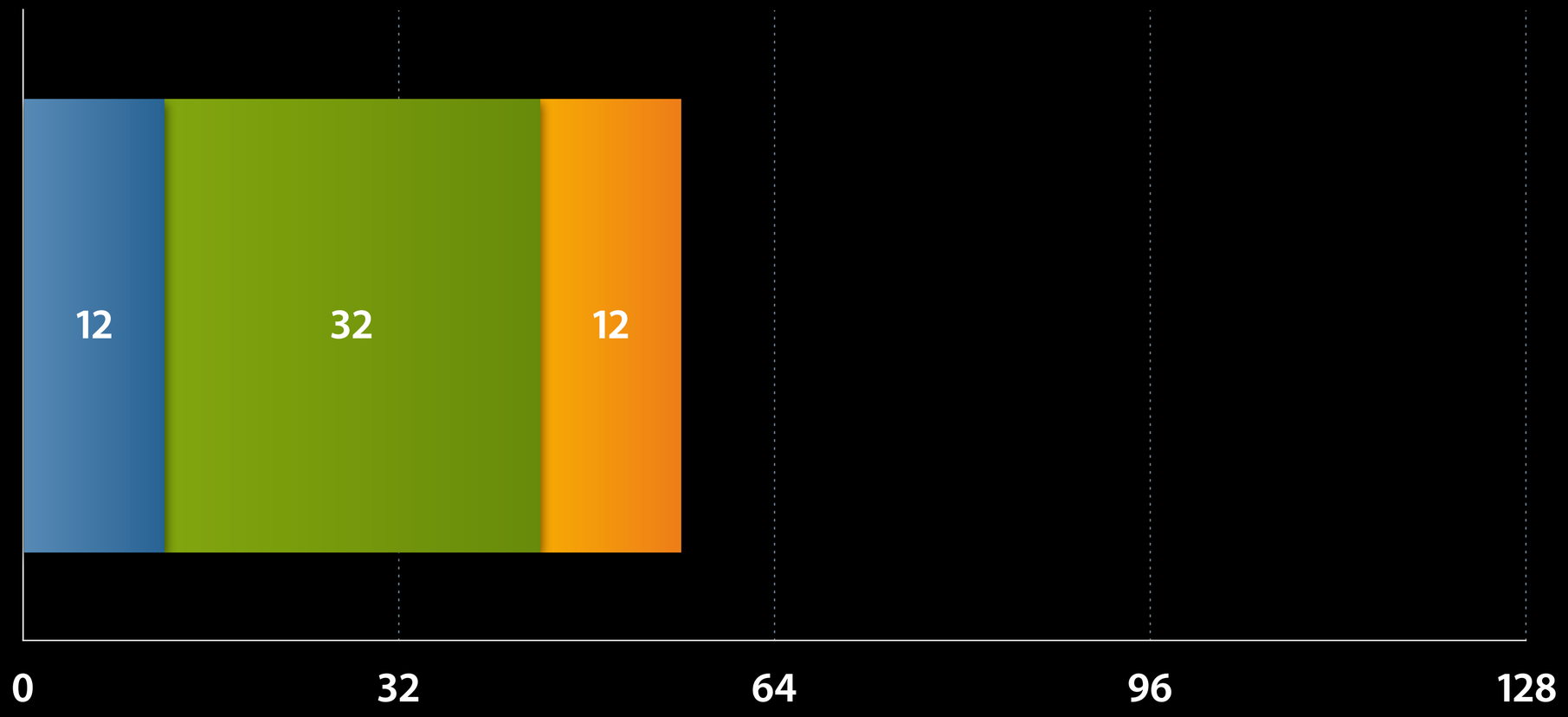
Memory Overview



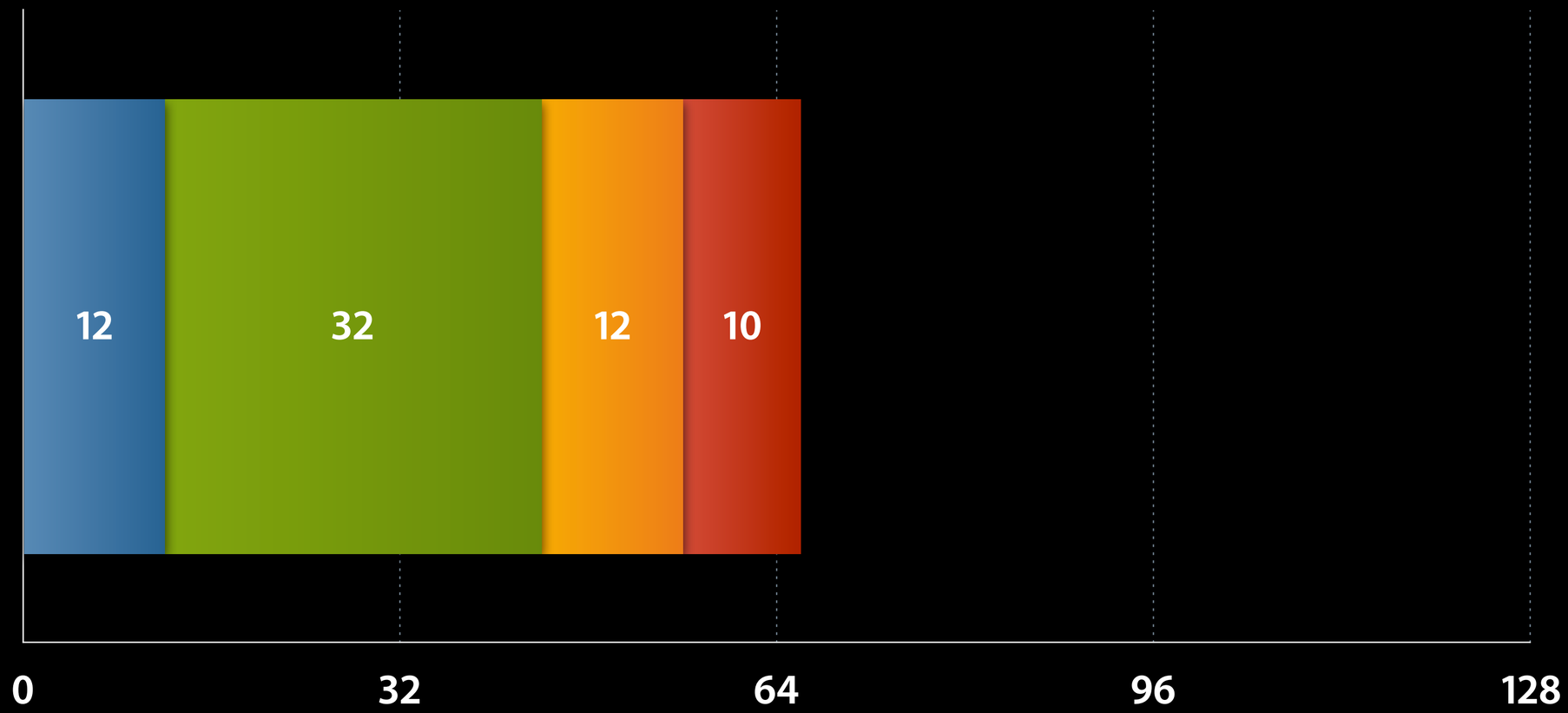
Memory Overview



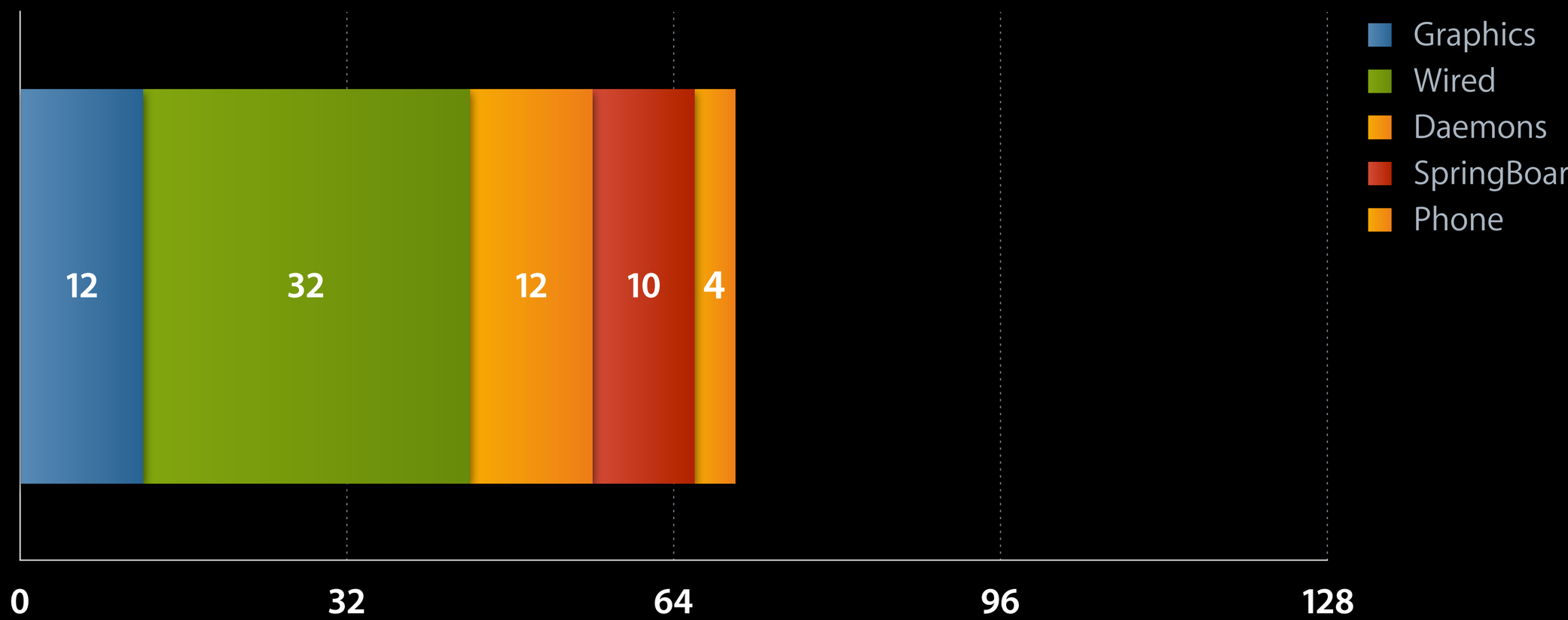
Memory Overview



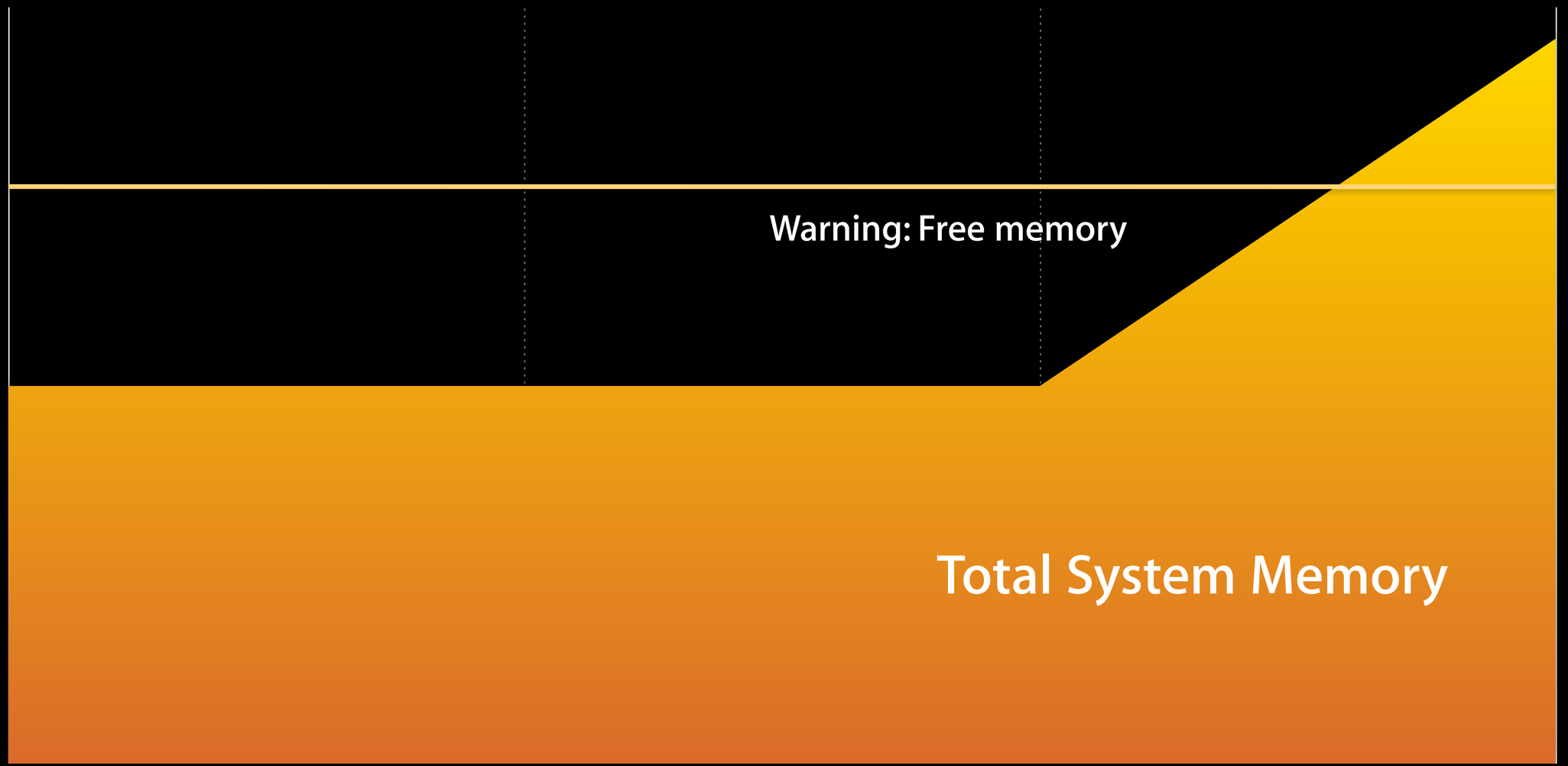
Memory Overview



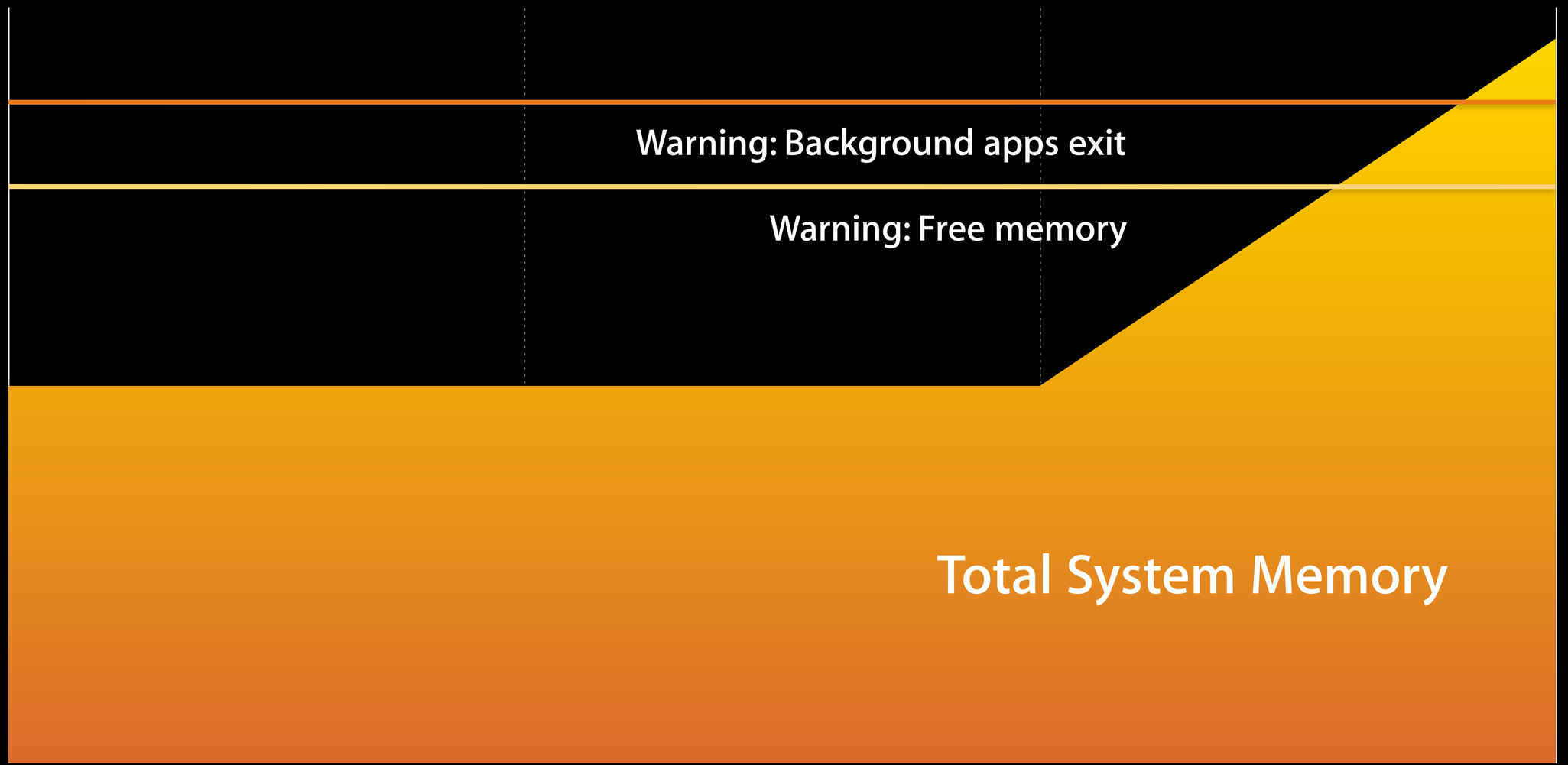
Memory Overview



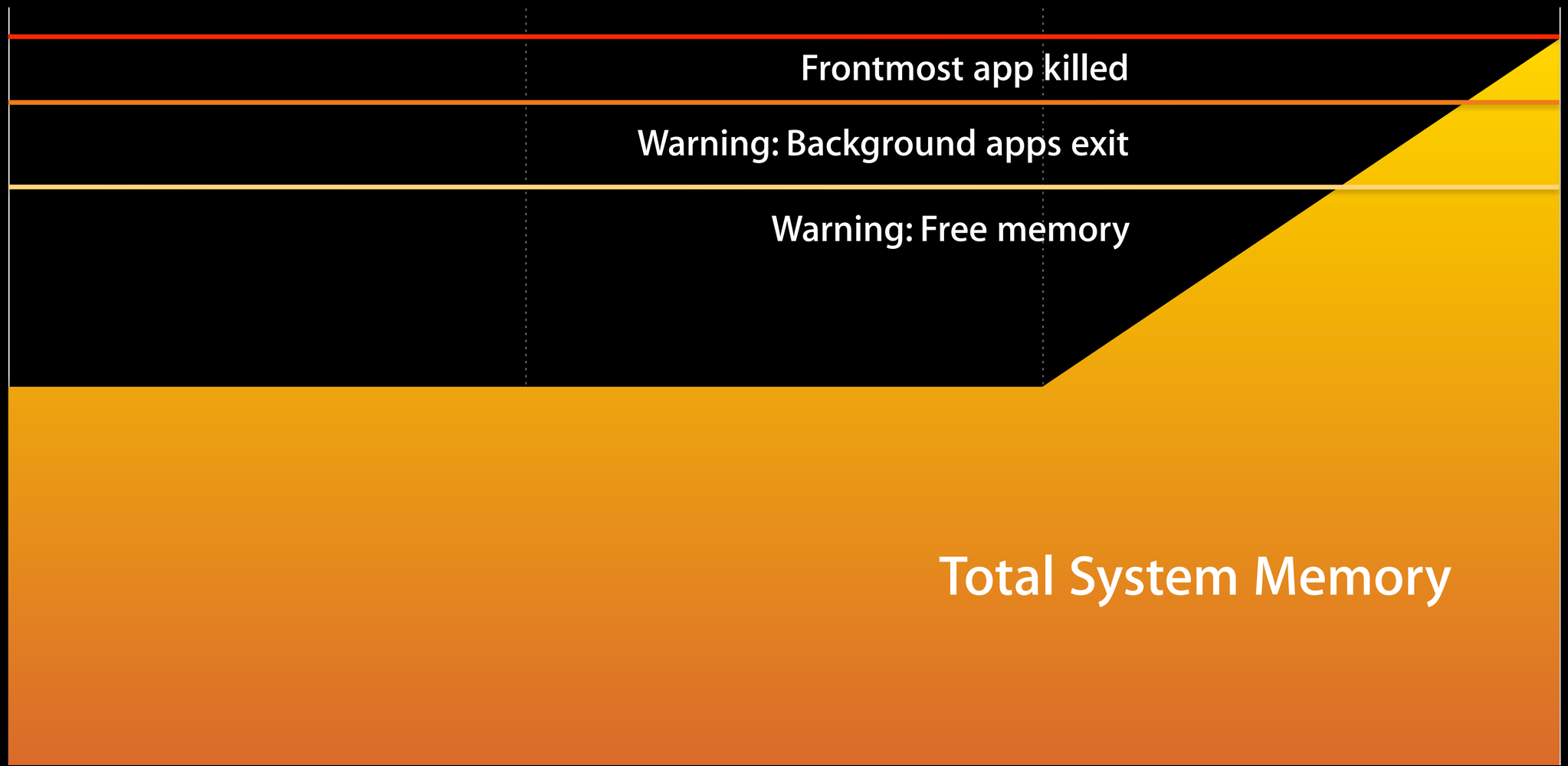
Memory Overview



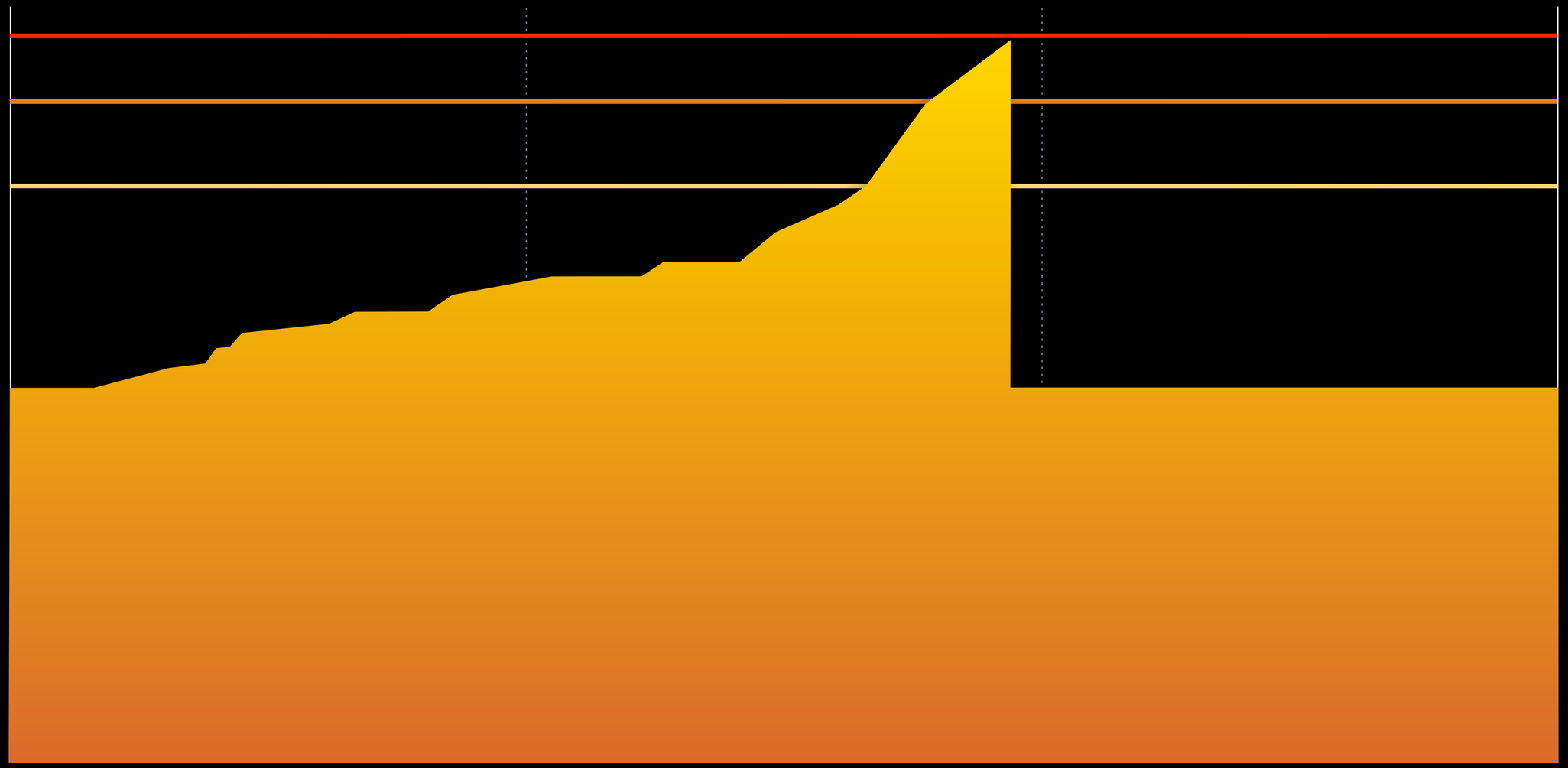
Memory Overview



Memory Overview



Memory Overview



Cocoa Memory Management

Managed memory model

- Objective-C
  - alloc/init to create objects
  - retain/release/autorelease to manage reference counts
- No garbage collection
- Standard C, C++ memory management

Dynamic Memory

Allocate wisely

- Cache the right objects
  - Cache frequently used objects
  - But, be prepared to release those objects if needed
- Free objects when you are done with them
- Monitor object allocations with ObjectAlloc instrument

Demo

Breadcrumbs app

Charles Srisuwananukorn
iPhone Performance

What You'll Learn

- Memory architecture on iPhone
- Memory tools
- Image and layer memory
- Responding to low memory warnings
- Autorelease pools
- Understanding low memory logs

Memory Usage Instruments

Dynamic Memory

Memory Leaks

Xcode Static Analyzer

- Analyzes source code
  - Finds memory bugs (and other bugs) without running the application
  - Enforces Cocoa memory management policy

Demo

Memory tools

Charles Srisuwananukorn
iPhone Performance

What You'll Learn

- Memory architecture on iPhone
- Memory tools
- Image and layer memory
- Responding to low memory warnings
- Autorelease pools
- Understanding low memory logs

Dynamic Memory

Image and layer memory

- Image and layer objects are small by themselves
- …but backing store is large
  - Uncompressed Core Graphics memory
  - Layer backing store shared between app and SpringBoard

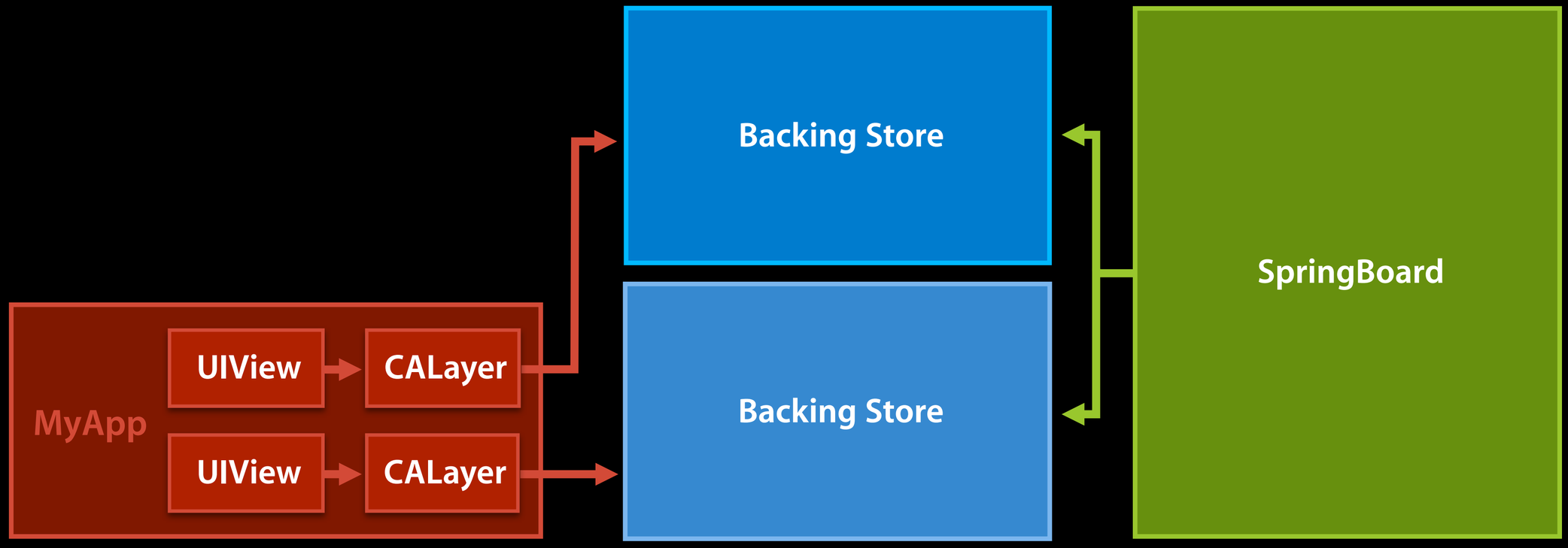
- Every UIView has a CALayer

```
UIKIT_EXTERN_CLASS @interface UIView : UIResponder<NSCoding> {
  @package
    CALayer *_layer;
...
```

Dynamic Memory

Layer memory

- Layer objects are small
- …but backing store is large, shared with SpringBoard



Dynamic Memory

Image and layer memory

- Images are decompressed in memory to display them
  - 320px × 480px × 4B/px = 614.4KB
  - 1600px × 1200px × 4B/px = 7.68MB
  - 10MP × 4B/px = 40MB

Dynamic Memory

Image and layer memory

- `+[UIImage imageNamed:]`
  - Caches uncompressed images in memory
  - Used by `-initWithNibName:`
    - Load large images using other initializers in `-viewDidLoad` instead
  - Clears the cache on a memory warning
    - In 3.0 it's more aggressive about releasing memory on a warning
  - Use for small, frequently used images

Dynamic Memory

Image and layer memory

- `-[UIImage initWithData:]`
  - Retains the NSData
  - On a memory warning, release the NSData if you can recreate it
  - Use `-[NSData initWithContentsOfMappedFile:]` for large files

Dynamic Memory

Image and layer memory

- `-[UIImage initWithContentsOfFile:]`
  - Only retains the path

Dynamic Memory

Image and layer memory

API	Caching	On Memory Warnings
`+[UIImage imageNamed:]`	Caches uncompressed images	Release any references; Load large images using other initializers
`-[UIImage initWithData:]`	Caches compressed NSData	Release NSData if possible
`-[UIImage initWithContentsOfFile:]`	Does not cache	Release if possible

Demo

Releasing large images

What You'll Learn

- Memory architecture on iPhone
- Memory tools
- Image and layer memory
- Responding to low memory warnings
- Autorelease pools
- Understanding low memory logs

Low Memory Warnings

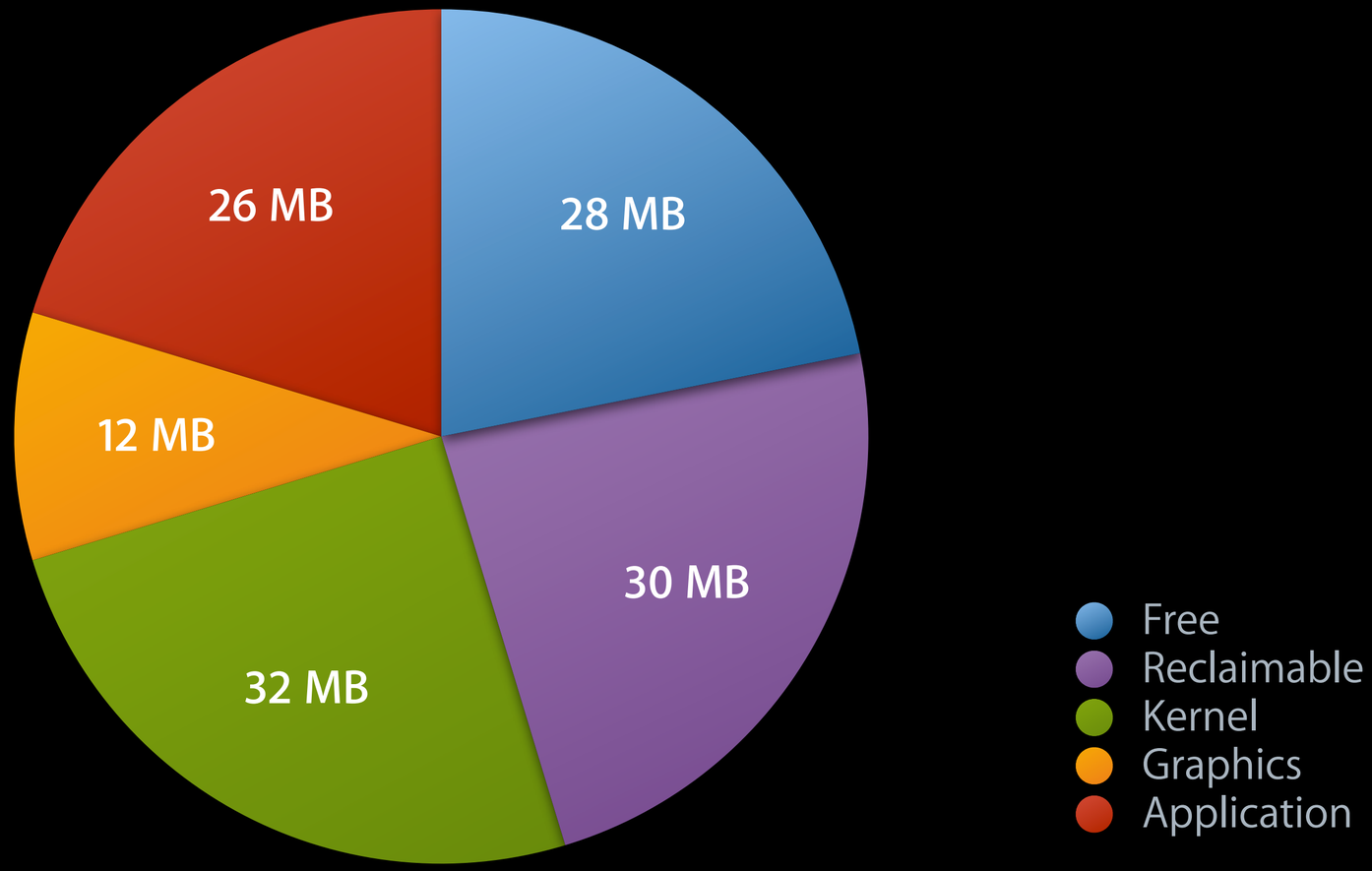
Overview

- One foreground app, many processes
- Each process consumes memory

Process ID	Process Name	User Name	% CPU	Threads	Real Mem	Virtual Mem	Architecture	CPU Time	Sudden Term
1	launchd	root	0	3	428.00 KB	12.74 MB	ARM	09.77	N/A
13	syslogd	root	0	4	520.00 KB	12.70 MB	ARM	06.75	N/A
14	lockdownd	root	0	4	1.34 MB	27.27 MB	ARM	1:41.93	N/A
15	mediaserverd	mobile	0	13	1.56 MB	20.59 MB	ARM	06.44	N/A
16	mDNSRespo...	_mdnsresp...	3.8	2	948.00 KB	11.76 MB	ARM	03.88	N/A
18	IQAgent	root	0	3	1.15 MB	13.42 MB	ARM	03.22	N/A
19	fairplayd	root	0	1	604.00 KB	12.05 MB	ARM	00.33	N/A
20	configd	root	0	9	1.62 MB	15.72 MB	ARM	33.58	N/A
21	configd	root	0	1	388.00 KB	10.73 MB	ARM	09.22	N/A
22	accessoryd	mobile	0	1	376.00 KB	10.54 MB	ARM	04.33	N/A
23	SpringBoard	mobile	0	9	7.87 MB	72.45 MB	ARM	3:58.39	N/A
27	CommCenter	root	0	9	1.43 MB	39.77 MB	ARM	31.43	N/A
28	BTServer	mobile	0	3	624.00 KB	15.07 MB	ARM	09.02	N/A
29	notifyd	root	0	2	292.00 KB	11.08 MB	ARM	10.47	N/A
446	MobilePhone	mobile	0	1	3.41 MB	61.52 MB	ARM	04.06	N/A
519	ptpd	mobile	0	2	1.36 MB	24.64 MB	ARM	00.41	N/A
543	notification_pro	root	0	2	668.00 KB	10.54 MB	ARM	00.04	N/A
547	DTMobileIS	root	7.6	5	6.69 MB	17.62 MB	ARM	05.19	N/A
550	installd	mobile	0	1	1.39 MB	11.57 MB	ARM	00.33	N/A

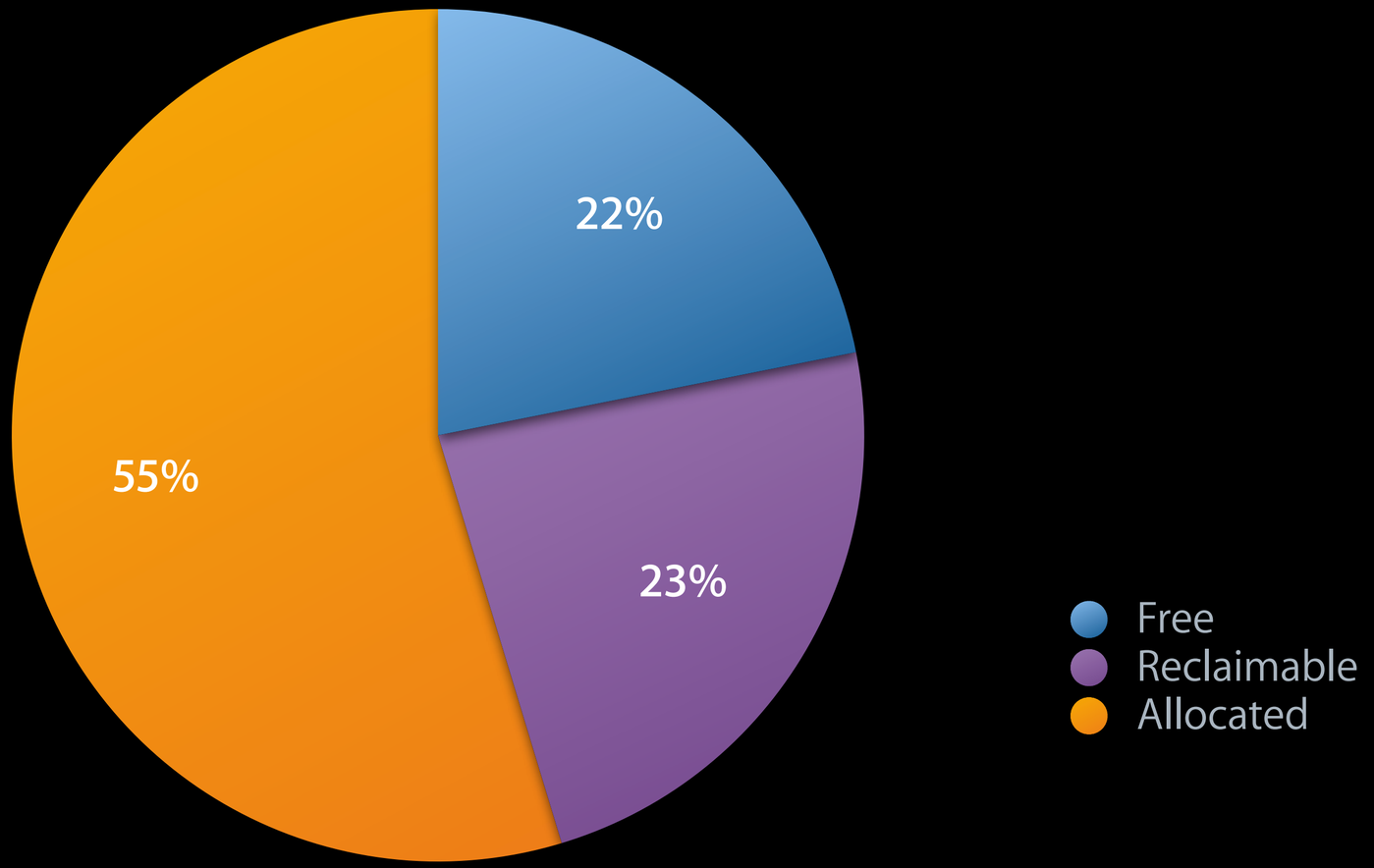
Low Memory Warnings

Overview



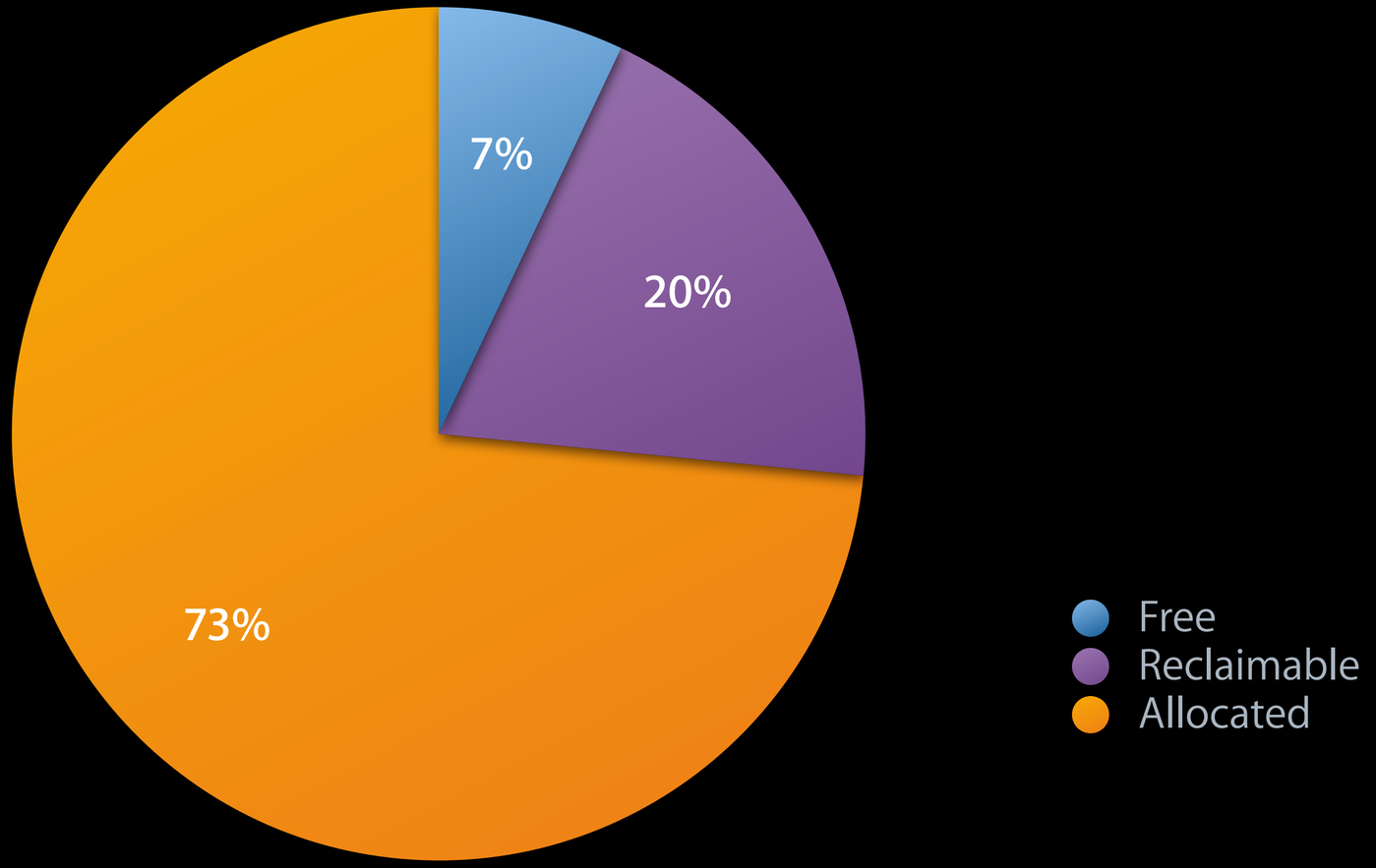
Low Memory Warnings

Overview



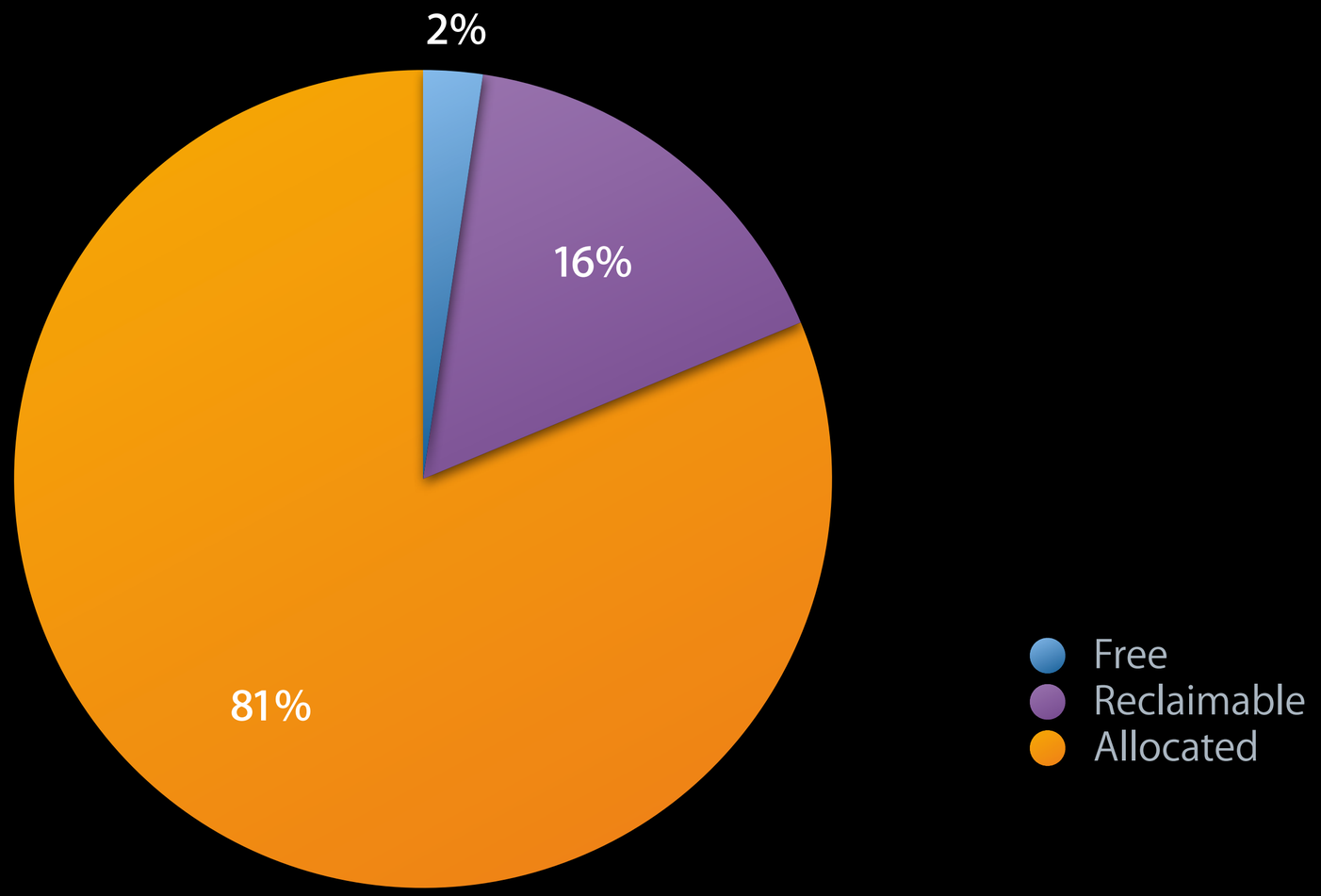
Low Memory Warnings

Overview



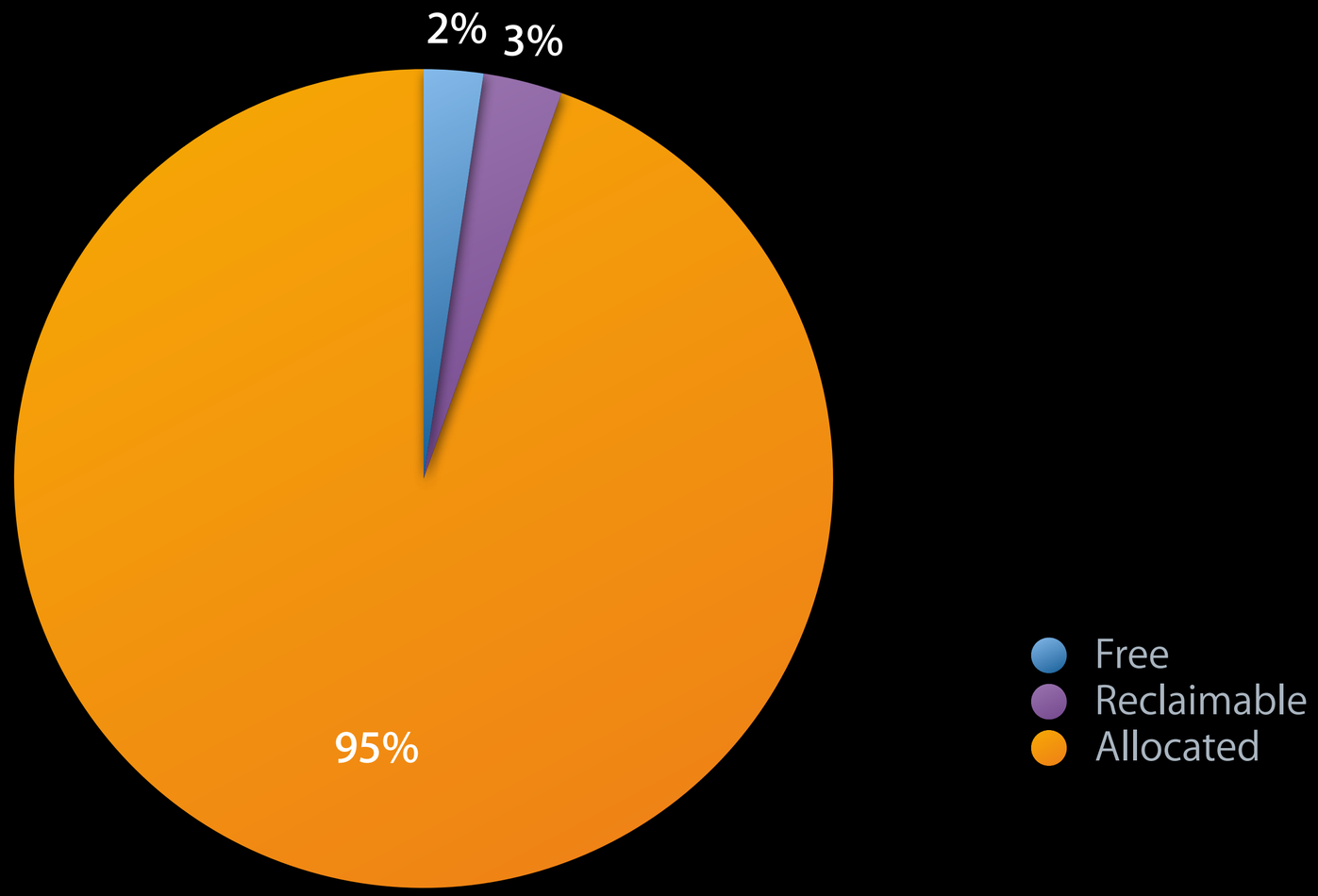
Low Memory Warnings

Overview

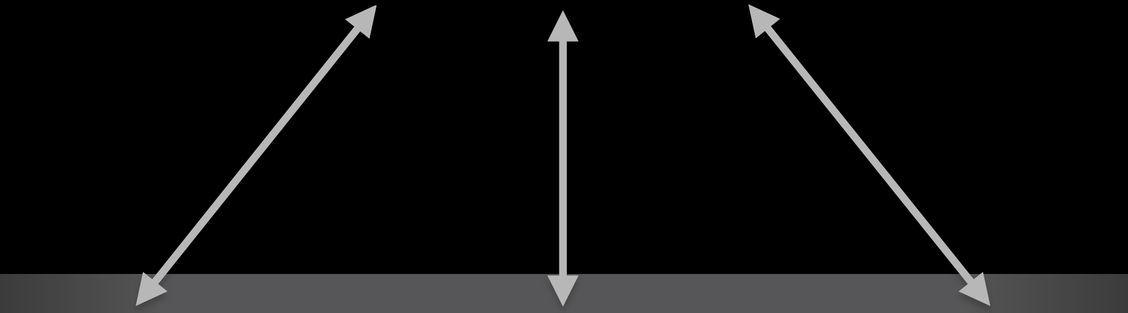


Low Memory Warnings

Overview



Low Memory Warnings



Safari

MyApplication

Mail

Low Memory Warnings

Don’t ignore the signs

- Any app may create low memory conditions
- Expect memory warnings (normal part of the system)
- You must respond to low memory warnings!
  - Failure to respond can cause app termination

Low Memory Warnings

Taking action

- Release any objects that can be reconstructed
- Release cached objects
- Unload cached resource files

Low Memory Warnings

Taking action

- Don’t ask user to do anything (they can’t!)

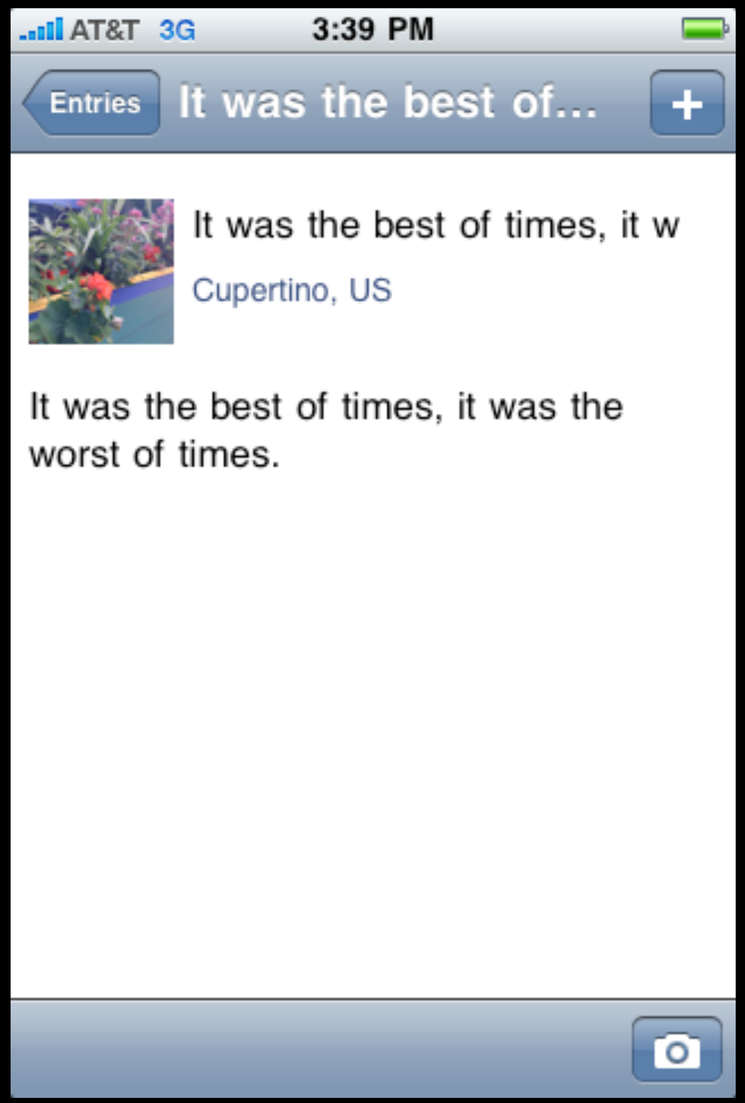
Low Memory Warnings

Unloading views

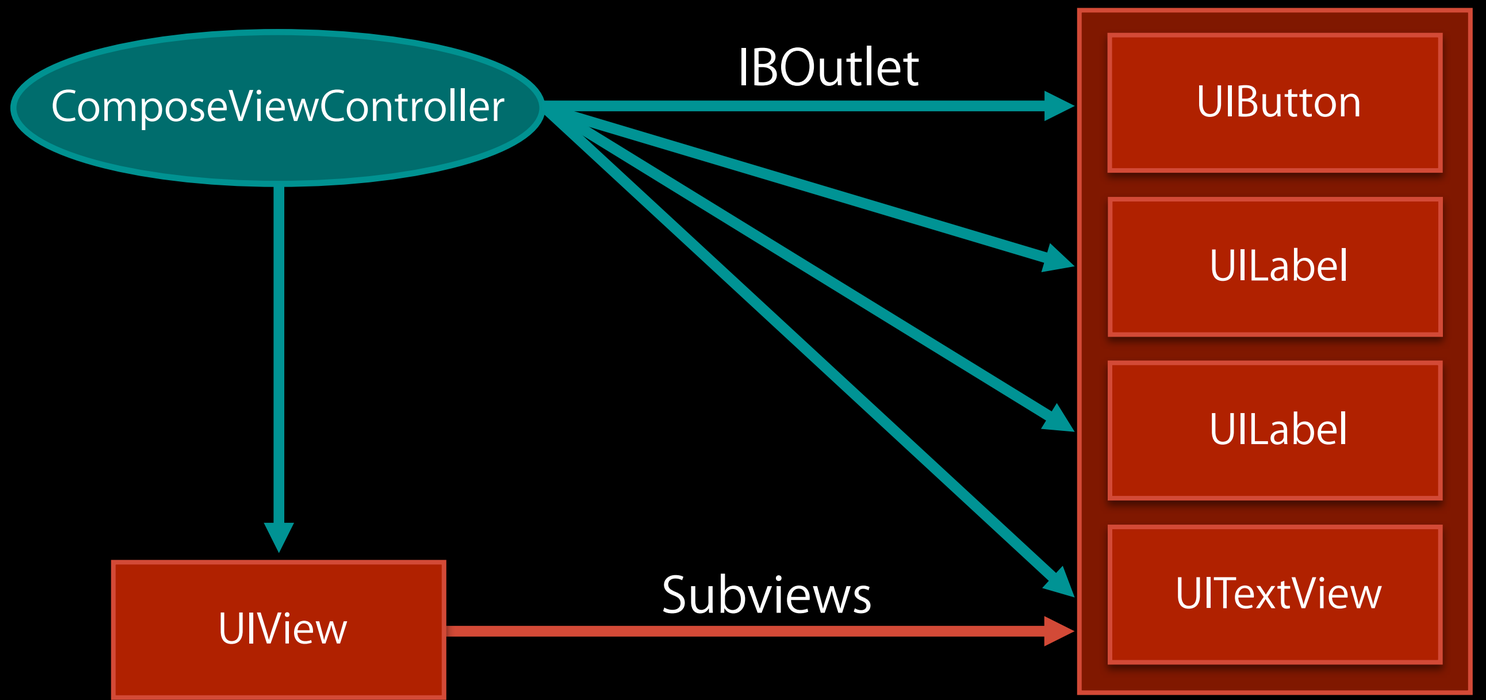
- `-[UIViewController didReceiveMemoryWarning]` may release offscreen views
- But it needs help releasing outlets

Low Memory Warnings

Unloading views



Low Memory Warnings



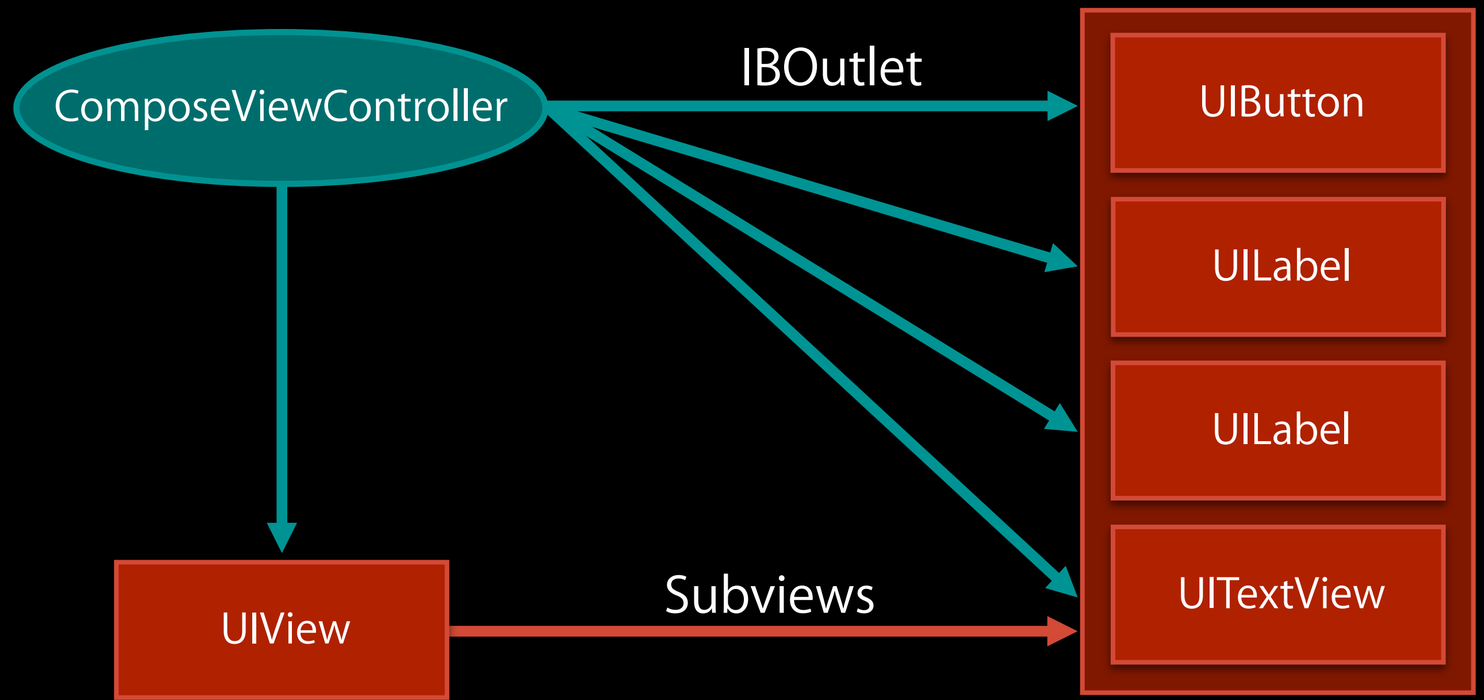
Low Memory Warnings

Unloading views

```
- (void)viewDidUnload {
    self.titleLabel = nil;
    self.locationLabel = nil;
    self.textView = nil;
    self.imageButton = nil;

    [super viewDidUnload];
}
```

Low Memory Warnings



Low Memory Warnings

Unloading views

- We've made this much easier in 3.0
- See "Memory Warnings" in the *Memory Management Programming Guide for Cocoa* for more details

Low Memory Warnings

Receiving low memory warnings

- In your app delegate
  - Implement `applicationDidReceiveMemoryWarning:` method
- In `UIViewController` subclasses
  - Override `didReceiveMemoryWarning` method
- Direct notifications
  - Register for `UIApplicationDidReceiveMemoryWarningNotification`

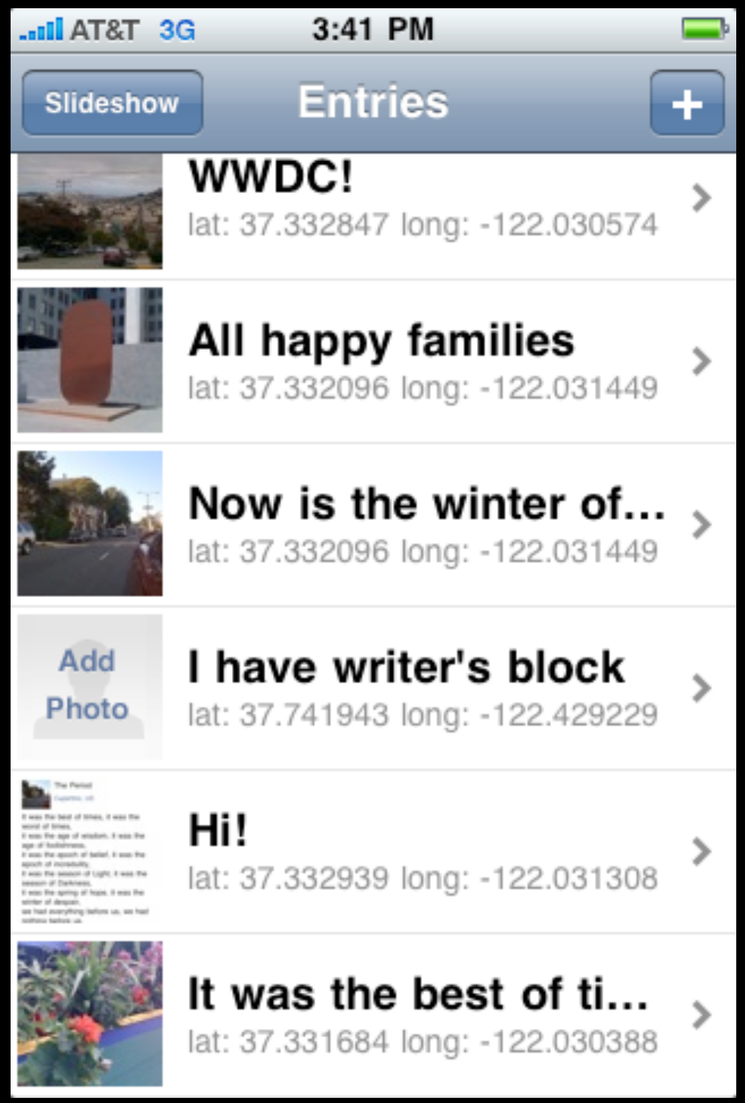
Low Memory Warnings

Receiving low memory warnings

- In your app delegate
  - Implement `applicationDidReceiveMemoryWarning:` method
- In `UIViewController` subclasses
  - Override `didReceiveMemoryWarning` method
- Direct notifications
  - Register for `UIApplicationDidReceiveMemoryWarningNotification`

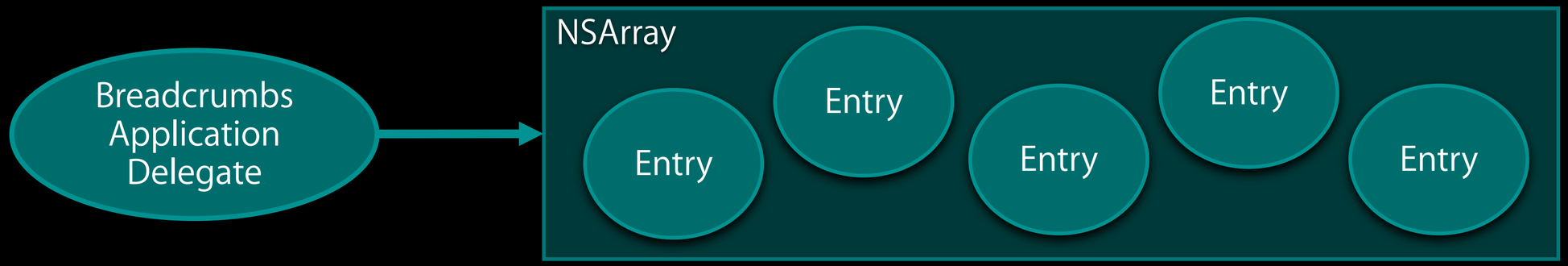
Low Memory Warnings

Receiving low memory warnings



Low Memory Warnings

Receiving low memory warnings



Low Memory Warnings

Receiving low memory warnings

```
- (NSArray *)entries {
    if (!m_entries) {
        BreadcrumbStore *breadcrumbStore = [BreadcrumbStore sharedInstance];
        m_entries = [[breadcrumbStore listBreadcrumbs] retain];
    }
    return m_entries;
}
```

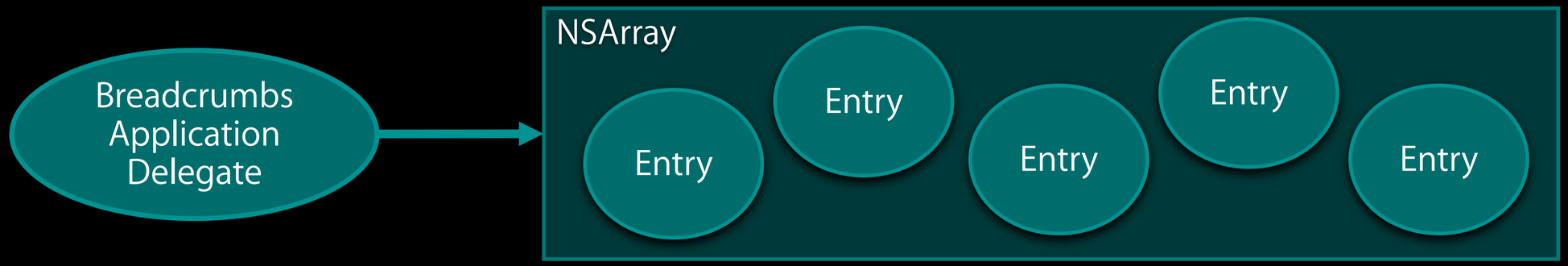
Low Memory Warnings

Receiving low memory warnings

```
- (void)applicationDidReceiveMemoryWarning:(UIApplication *)application {
    [m_entries release];
    m_entries = nil;
}
```

Low Memory Warnings

Receiving low memory warnings



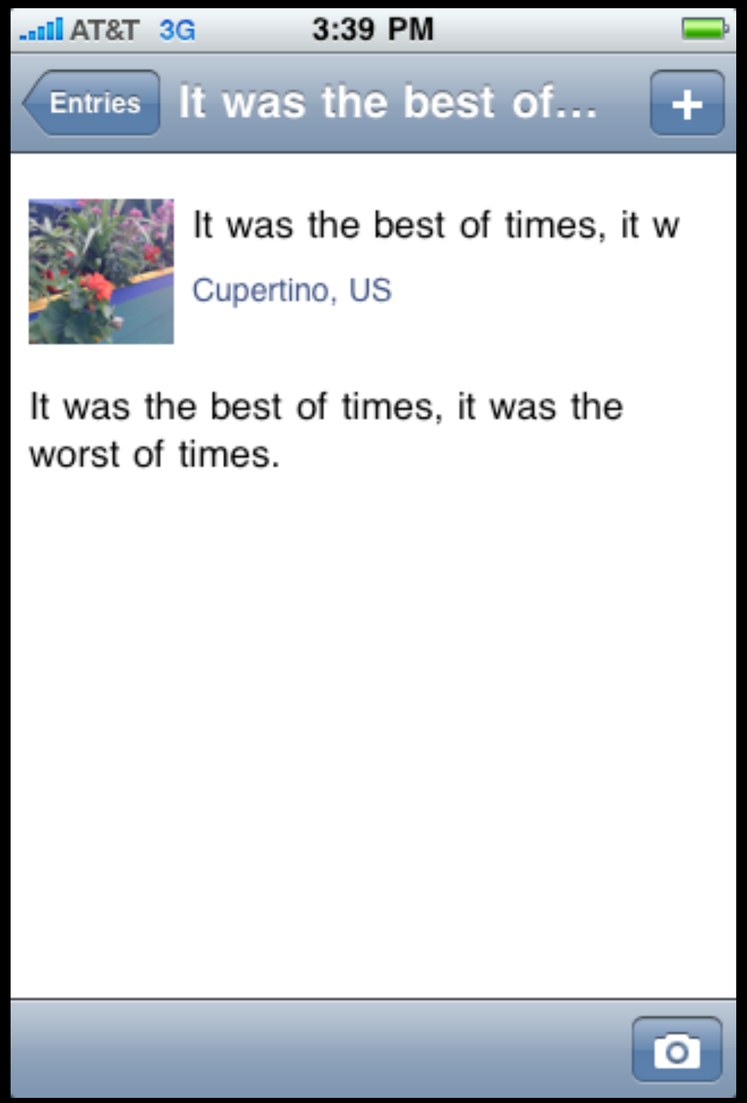
Low Memory Warnings

Receiving low memory warnings

- In your app delegate
  - Implement `applicationDidReceiveMemoryWarning:` method
- In `UIViewController` subclasses
  - Override `didReceiveMemoryWarning` method
- Direct notifications
  - Register for `UIApplicationDidReceiveMemoryWarningNotification`

Low Memory Warnings

Receiving low memory warnings



Low Memory Warnings

Receiving low memory warnings

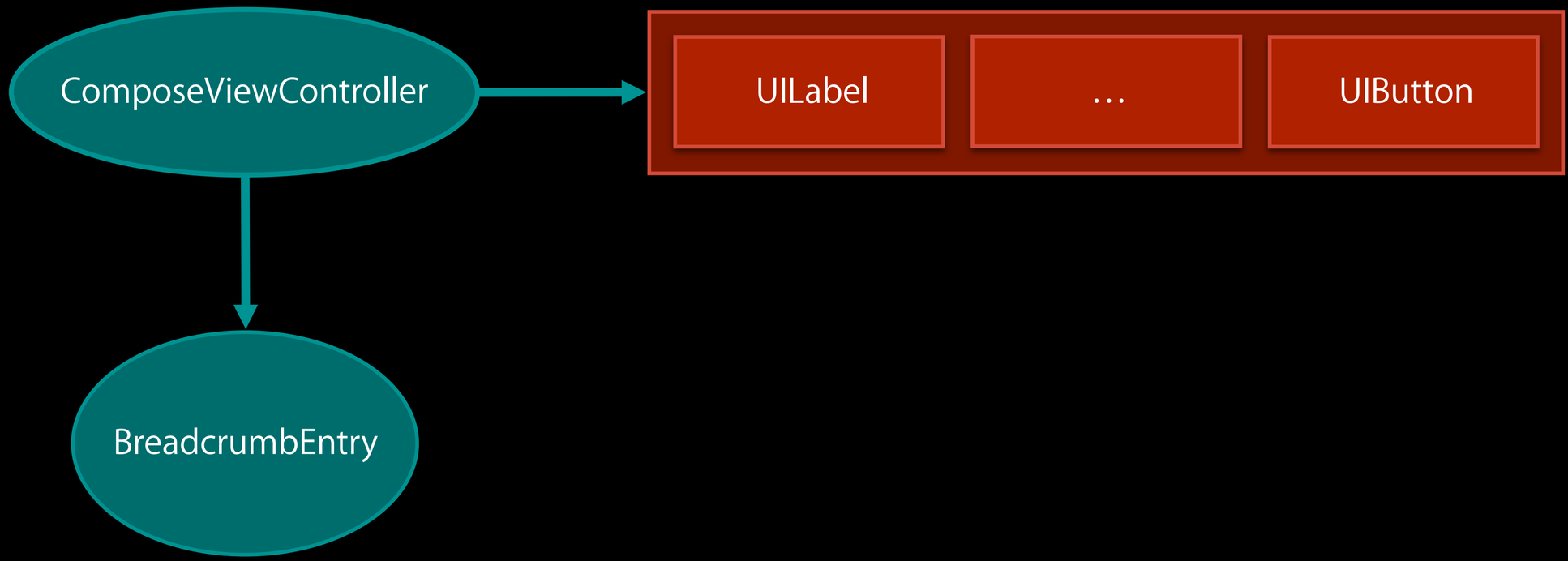
```
@interface ComposeViewController : /* ... */ {
    // ...
}

@property (nonatomic, retain) IBOutlet UILabel *titleLabel;
@property (nonatomic, retain) IBOutlet UILabel *locationLabel;
@property (nonatomic, retain) IBOutlet UITextView *textView;
@property (nonatomic, retain) IBOutlet UIButton *imageButton;
@property (nonatomic, retain) BreadcrumbEntry *entry;

// ...

@end
```

Low Memory Warnings



Low Memory Warnings

Receiving low memory warnings

```
@interface BreadcrumbEntry : NSObject {
    // ...
}

@property (nonatomic, retain) NSString *summary;
@property (nonatomic, retain) UIImage *thumbnail;
// ...
```

```
@property (nonatomic, retain) NSString *text;
@property (nonatomic, retain) UIImage *image;
// ...
```

```
@property (nonatomic, assign) BOOL imageDirty;
// ...
```

```
- (void)receivedMemoryWarning;
```

```
@end
```

Low Memory Warnings

Receiving low memory warnings

```
- (void)didReceiveMemoryWarning {
    [m_entry receivedMemoryWarning];

    [super didReceiveMemoryWarning];
}
```

Low Memory Warnings

Receiving low memory warnings

- In your app delegate
  - Override `applicationDidReceiveMemoryWarning:` method
- In `UIViewController` subclasses
  - Override `didReceiveMemoryWarning` method
- Direct notifications
  - Register for `UIApplicationDidReceiveMemoryWarningNotification`

Low Memory Warnings

Receiving low memory warnings

```
static BreadcrumbStore *sharedStore = nil;

// ...

+ (id)sharedInstance {
    if (!sharedStore) {
        sharedStore = [[BreadcrumbStore alloc] init];
        NSNotificationCenter *center = [NSNotificationCenter defaultCenter];
        [center addObserver:[BreadcrumbStore class]
                   selector:@selector(receivedMemoryWarning)
                       name:UIApplicationDidReceiveMemoryWarningNotification
                     object:nil];
    }
    return sharedStore;
}
```

Low Memory Warnings

Receiving low memory warnings

```
+ (void)receivedMemoryWarning {
    [sharedStore release];
    sharedStore = nil;

    NSNotificationCenter *defaultCenter = [NSNotificationCenter defaultCenter];
    [defaultCenter removeObserver:[BreadcrumbStore class]];
}
```

Demo

Memory warnings in the simulator

What You'll Learn

- Memory architecture on iPhone
- Memory tools
- Image and layer memory
- Responding to low memory warnings
- Autorelease pools
- Understanding low memory logs

Autorelease Pools

Event loop

```
UIApplicationMain
  -[UIApplication _run]
     GSEventRunModal
       CFRunLoopRunInMode
         CFRunLoopRunSpecific
           PurpleEventCallback
        _UIApplicationHandleEvent
           NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
             /* UIKit and your app handle the event */
           [pool drain];
```

Autorelease Pools

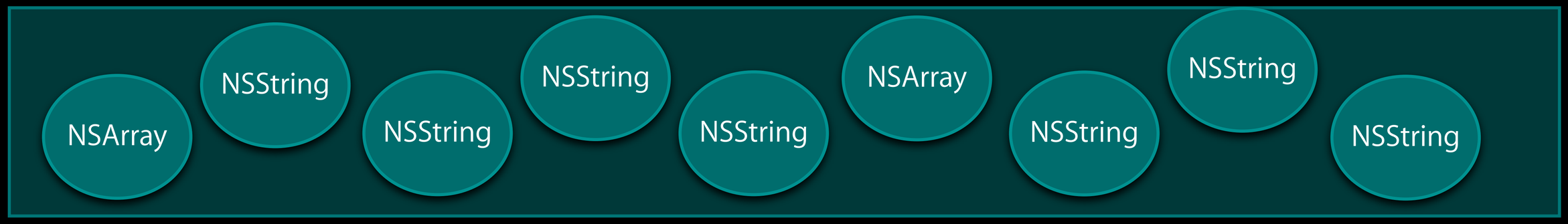
Nested autorelease pools

```
while (entry = [self nextEntryFromDatabase]) {
      NSCharacterSet *newlineCharacterSet = [NSCharacterSet newlineCharacterSet];
      NSArray *lines = [entry componentsSeparatedByString:newlineCharacterSet];
      for (NSString *line in lines) {
         NSCharacterSet *charSet = [NSCharacterSet whitespaceAndNewlineCharacterSet];
         NSString *trimmedLine = [line stringByTrimmingCharactersInSet:charSet];
      }
}
```

Autorelease Pools

Nested autorelease pools

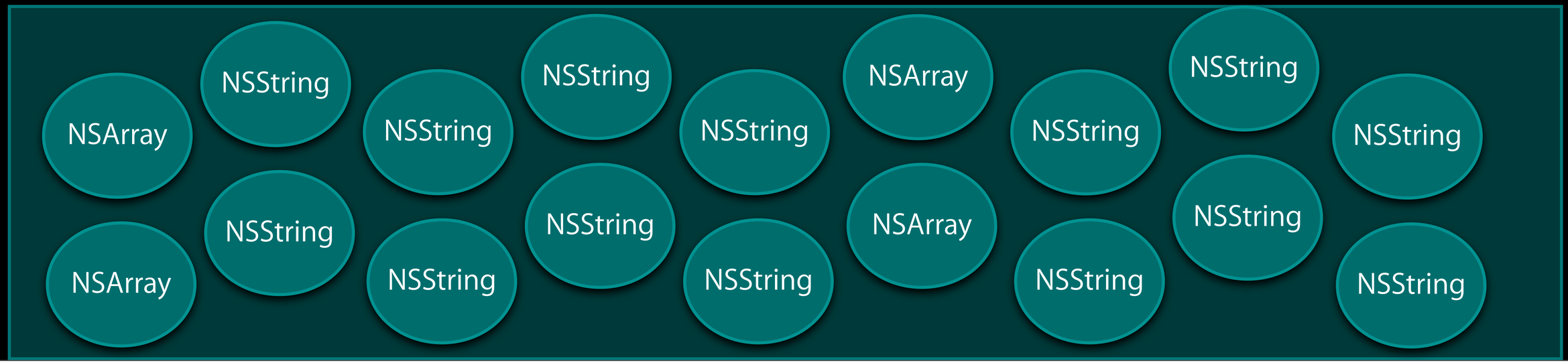
```
while (entry = [self nextEntryFromDatabase]) {
      NSCharacterSet *newlineCharacterSet = [NSCharacterSet newlineCharacterSet];
      NSArray *lines = [entry componentsSeparatedByString:newlineCharacterSet];
      for (NSString *line in lines) {
         NSCharacterSet *charSet = [NSCharacterSet whitespaceAndNewlineCharacterSet];
         NSString *trimmedLine = [line stringByTrimmingCharactersInSet:charSet];
      }
}
```



Autorelease Pools

Nested autorelease pools

```
while (entry = [self nextEntryFromDatabase]) {
      NSCharacterSet *newlineCharacterSet = [NSCharacterSet newlineCharacterSet];
      NSArray *lines = [entry componentsSeparatedByString:newlineCharacterSet];
      for (NSString *line in lines) {
         NSCharacterSet *charSet = [NSCharacterSet whitespaceAndNewlineCharacterSet];
         NSString *trimmedLine = [line stringByTrimmingCharactersInSet:charSet];
      }
}
```



Autorelease Pools

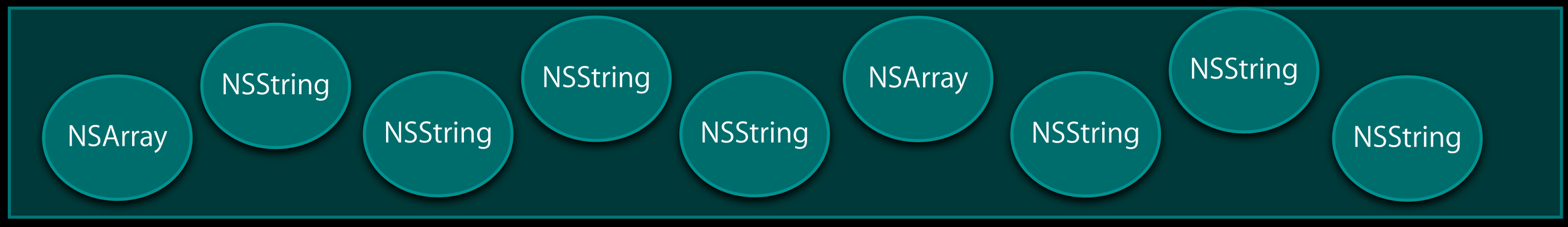
Nested autorelease pools

```
while (entry = [self nextEntryFromDatabase]) {
      NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
      NSCharacterSet *newlineCharacterSet = [NSCharacterSet newlineCharacterSet];
      NSArray *lines = [entry componentsSeparatedByString:newlineCharacterSet];
      for (NSString *line in lines) {
          NSCharacterSet *charSet = [NSCharacterSet whitespaceAndNewlineCharacterSet];
          NSString *trimmedLine = [line stringByTrimmingCharactersInSet:charSet];
      }
      [pool drain];
}
```

Autorelease Pools

Nested autorelease pools

```
while (entry = [self nextEntryFromDatabase]) {
      NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
      NSCharacterSet *newlineCharacterSet = [NSCharacterSet newlineCharacterSet];
      NSArray *lines = [entry componentsSeparatedByString:newlineCharacterSet];
      for (NSString *line in lines) {
          NSCharacterSet *charSet = [NSCharacterSet whitespaceAndNewlineCharacterSet];
          NSString *trimmedLine = [line stringByTrimmingCharactersInSet:charSet];
      }
      [pool drain];
}
```



Autorelease Pools

Nested autorelease pools

```
while (entry = [self nextEntryFromDatabase]) {
      NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
      NSCharacterSet *newlineCharacterSet = [NSCharacterSet newlineCharacterSet];
      NSArray *lines = [entry componentsSeparatedByString:newlineCharacterSet];
      for (NSString *line in lines) {
          NSCharacterSet *charSet = [NSCharacterSet whitespaceAndNewlineCharacterSet];
          NSString *trimmedLine = [line stringByTrimmingCharactersInSet:charSet];
      }
      [pool drain];
}
```

Dynamic Memory

Avoid autoreleasing objects

- Autorelease can be expensive
  - Use release whenever possible
- Use direct allocation instead of convenience methods

```
NSString* aString = [[NSString alloc] initWithFormat:@”Text”];

[myDictionary setObject:aString forKey:@”MyKey”];

[aString release];
```

instead of:

```
NSString* aString = [NSString stringWithFormat:@”Text”];

[myDictionary setObject:aString forKey:@”MyKey”];
```

Demo

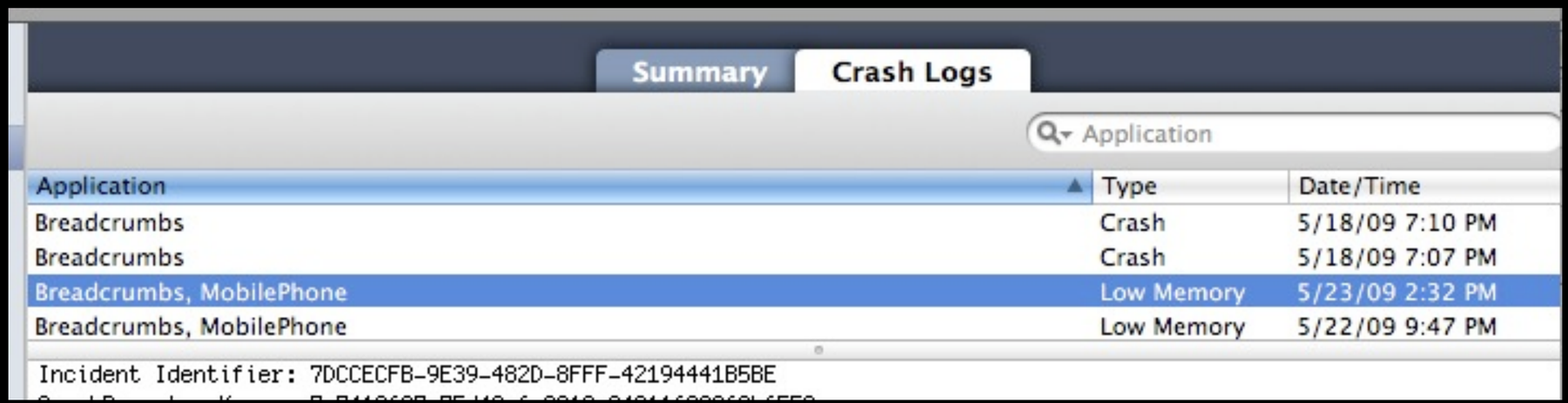
Nested autorelease pools

What You'll Learn

- Memory architecture on iPhone
- Memory tools
- Image and layer memory
- Responding to low memory warnings
- Autorelease pools
- Understanding low memory logs

Low Memory Logs

- Low-Memory crash logs show up in Xcode Organizer under the “Crash Logs” tab



Low Memory Logs

```
Incident Identifier: 4EA7F240-B90C-4EA6-A7F7-0CC4329CCEEF
CrashReporter Key:   7a7412607a75d49efa9919e343116020f9b6553c
OS Version:          iPhone OS 3.0 (7A341)
Date:                2009-06-01 21:36:43 -0700

Free pages:        372
Wired pages:       8678
Purgeable pages:   0
Largest process:   Breadcrumbs

Processes
          Name                UUID                    Count resident pages
       locationd <777d750ce6d3eb590afe23e43eb7a7d9>     255
     Breadcrumbs <6cff3b7a1e48ef88c23fa7ab9bd7a117>   16196 (jettisoned) (active)
       DTMobileIS <a541ccb799a4e87cb10636268ab1b87b>    1805
notification_pro <09df4bf987a6baaa3e77192c4026bc5e>      58
    syslog_relay <4adc9d69d534cb0be0649055b2188c4e>      57
notification_pro <09df4bf987a6baaa3e77192c4026bc5e>      59
             ptpd <e1c6f82741b354c1086e373473131824>     134
     MobilePhone <9517d3db1b487371c5a64e41bfd19eda>     429 (jettisoned)
         notifyd <69e04d13344f1c006227422dbf18dde8>      68
        BTServer <3ae1c6fbcb29741b4a609b02f3c5a127>     139
      CommCenter <3bb4f68851d96e651f3e637a13c5a50e>     294
     SpringBoard <e0166bb2d42cbc489ace07849c959088>    1415 (active)
      accessoryd <0a9234886eabf7a79e1cb822a7d2af81>      95
         configd <2950435c092830011e402b4ac8ea86fe>      74
         configd <2950435c092830011e402b4ac8ea86fe>     303
       fairplayd <ec3c2c10efbb2e7a7b9d7130ecb1e2c6>      71
         IQAgent <0c8ec49a509e684ffd4cae7729e2dcc9>     117
   mDNSResponder <21273afce237d1ff8e79ab819a130e3c>      86
    mediaserverd <cc588f963f30bd34296bf8beeb803f8a>     227
       lockdownd <374e220ffb0f06ac0f7aa024a979e31f>     256
         syslogd <4eae063fb35192e82cf5c47df1063443>      69
         launchd <09ab79ab1b390e1fee1309f58948f17c>      95

**End**
```

```
Free pages:          372
Wired pages:         8678
Purgeable pages:     0
Largest process:     Breadcrumbs
```

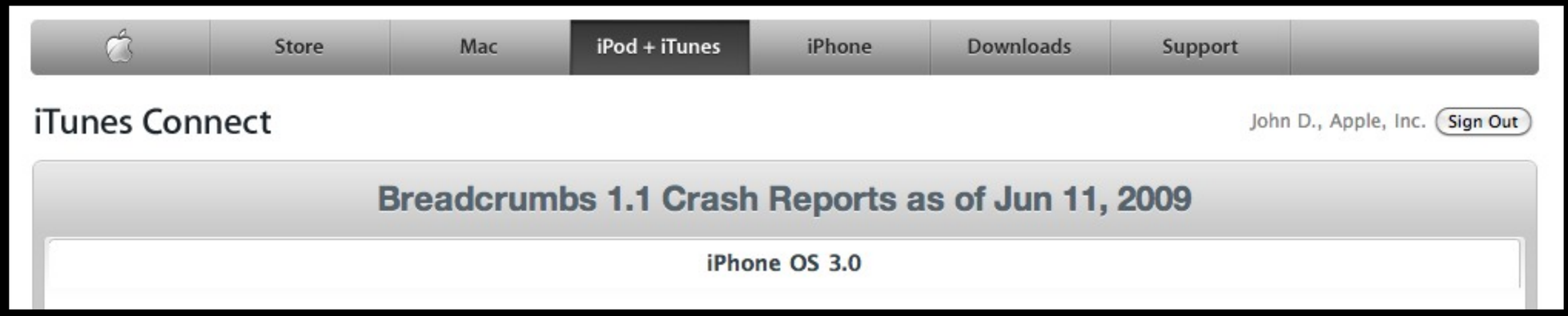
Name	UUID	Count resident pages
locationd	<777d750...>	255
Breadcrumbs	<6cff3b...>	16196 (jettisoned) (active)
SpringBoard	<e0166b...>	1415 (active)
accessoryd	<0a92348...>	95

Each page is 4 KB
16196 pages is equivalent to 64 MB.

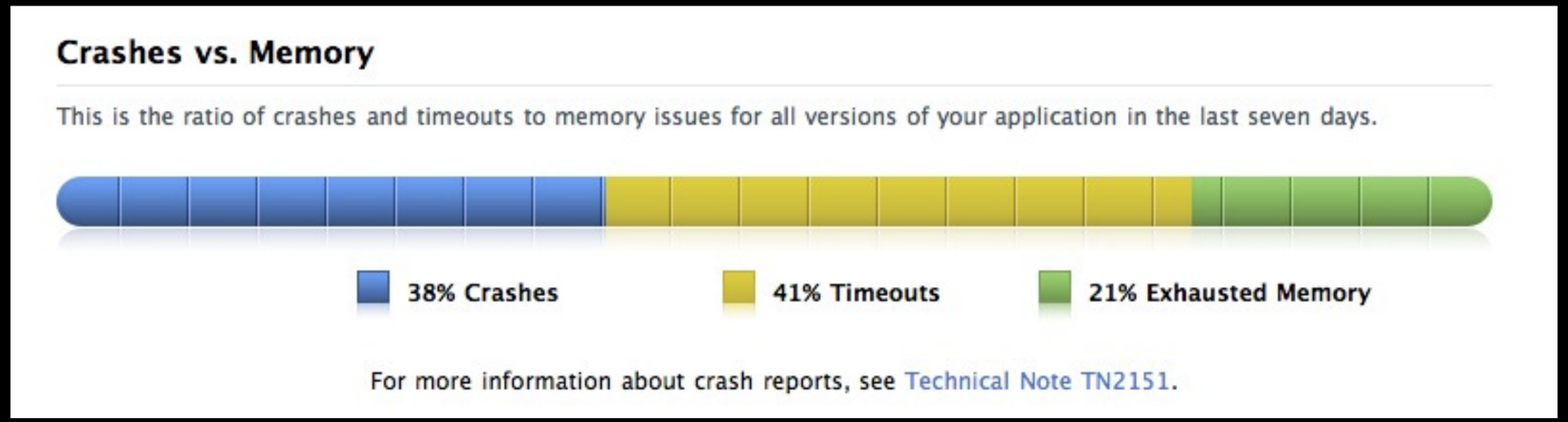
Low Memory Logs

- Things that may consume memory
  - MobileMail
  - MobilePhone
  - MobileSafari
  - mediaserverd
  - …
- File a bug if something other than your app and SpringBoard is the largest app

iTunes Connect



iTunes Connect



iTunes Connect

Memory

These are the memory issues for all versions of your application in the last seven days.

Resident Private Memory Size	
Average	20.25 MB
Maximum	24.8 MB

Summary

- Use ObjectAlloc, Leaks, Xcode Static Analyzer, and Activity Monitor to diagnose memory issues
- Follow Cocoa Memory Guidelines
- Avoid autorelease except at API boundaries
- Release images when they are no longer needed
- Respond to memory warnings

Related Sessions

Optimizing Performance on iPhone	Pacific Heights Thursday 10:30AM
iPhone Performance Optimization with Instruments	Marina Thursday 5:00PM
Performance Tuning with Shark on Mac and iPhone	North Beach Friday 9:00AM

Labs

iPhone Performance Lab	iPhone Lab A Tuesday 9:00AM
iPhone Memory Optimization Lab	iPhone Lab A Thursday 12:00PM
Performance Tuning with Shark on Mac and iPhone Lab	iPhone Lab C Friday 2:00PM
iPhone Performance Lab	iPhone Lab B Friday 9:00AM
iPhone Performance Optimization with Instruments Lab	iPhone Lab B Friday 2:00PM

More Information

Michael Jurewitz
Developer Tools Evangelist
jurewitz@apple.com

Documentation
Memory Management Programming Guide for Cocoa
http://developer.apple.com/iphone

Memory Usage Performance Guidelines
http://developer.apple.com/iphone

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09